Simulation-based Evaluation of Coin Selection Strategies

Murch
dev@murch.one

1 Motivation

Bitcoin's UTXO set, the set of all spendable balances in Bitcoin, has doubled in the past year and grown seven-fold in the past three years [4]. A large UTXO set is problematic since it causes an increased cost to operate each full node. Particularly, the UTXO set must be stored for operation and frequently searched for transaction verification. There are some on-going efforts to change the basic cost constraints of transactions such as Segregated Witness's discount for inputs [1] and Schnorr signature's promise to reduce transaction size in general as well as the possibility of collapsing signatures of multiple inputs [2].

However, a complementary approach is to reduce the UTXO set size by generally improving the *Coin Selection*, i.e. how wallets select inputs for transactions. The Coin Selection influences block space demand, fee costs, and composition of the wallet's available UTXOs. The proposed talk will present an analysis of the Coin Selection problem, a framework to simulate various Coin Selection approaches, and results of simulating the Coin Selection of multiple prominent Bitcoin wallets on a real-world scenario.

2 Simulation

The simulation framework implements simplified models for `Wallet`, `Transaction`, and `UTXO`. Payment activity is provided by the `Simulator` class that emits outgoing and incoming payment orders to the `Wallet` instances. Multiple classes inheriting from `Wallet` have been implemented with the Coin Selection strategies of Bitcoin Core, Bitcoin Wallet for Android (AndroidWallet), BreadWallet, and Mycelium. An overview of the strategies is provided in Table 1. Additionally, proposed approaches for improved Coin Selection are tested such as equi-random selection (RandomWallet) and aiming to create change of the same size as the spending target (DoubleWallet).

Wallet Type	Strategy
CoreWallet	Direct match, knapsack, or smallest larger UTXO. MIN_CHANGE of 0.01 BTC
MyceliumWallet	FIFO with pruning
BreadWallet	FIFO
AndroidWallet	By priority ($value \times inputAge$)
DoubleWallet	Core-style with MIN_CHANGE equal to spending target
RandomWallet	Equi-random UTXOs

Table 1: The implemented wallet strategies

The `Wallet` instances can be initialized with either an empty or a given set of available UTXOs. An incoming payment is added as a new `UTXO` to the wallet's available UTXOs. Outgoing payment requests trigger creation of a transaction in the wallet model. Hereby, the wallet performs Coin Selection for the requested amount. The Coin Selection abides to a fee per byte constraint. Finally, the selected transaction inputs are removed from the wallet model's available UTXOs. An optional change output is added to the wallet's available UTXOs exactly like an incoming payment.

A `UTXO` has a unique id, bitcoin value, and block height of first confirmation.

`Transaction` stores the selected input set, the target, the change, a fee, and also a block confirmation height.

Fee is estimated as a fixed cost per byte of 10 000 satoshi per kilobyte, i.e. 10 satoshi per byte. All transactions are assumed to be Pay-to-Public-Key-Hash transactions: 10 bytes for transaction overhead, 34 bytes per output, and 148 bytes per input. Simulation based on P2WPKH (SegWit) transaction sizes has also been performed.

We end up with a corpus of transactions for each wallet instance which we can use to derive statistical

Wallet Type	∅ UTXO	final UTXO	∅ change	max inputs	∅ inputs	total fees
CoreWallet	137.43	52	3 253 380.38	126	3.05	80 636 320
MyceliumWallet	774.30	1 097	17 044 311.52	166	2.91	61 749 175
BreadWallet	182.87	240	39 962 282.71	98	3.03	62 551 844
AndroidWallet	2 551.52	6 569	78 951 618.75	1 394	2.50	53 183 217
DoubleWallet	225.00	247	19 838 623.92	298	3.03	82 875 240
RandomWallet	182.17	258	38 375 535.64	87	3.03	62 536 067

Table 2: Performance of Coin Selection approaches on the MoneyPot.com scenario

data, and to look at the selection for specific payment requests. As the simulation framework adheres closely to the actual Bitcoin structure, we propose that it is useful to predict performance of Coin Selection strategies on the Bitcoin network. Transactions can have multiple inputs, but only one or two outputs. The framework doesn't include changing fees or mixed types of transactions, yet.

3 Scenarios

Ryan Havar has kindly provided transaction data from MoneyPot.com's hot wallet. This dataset contains an ordered set of 11 860 outgoing payments and 24 388 incoming payments [3]. Most incoming payments are at least 1 000 satoshi, 375 are below dustlimit. Outgoing payments are at least 0.1 mBTC (10 000 satoshi). The scenario is initialized with an empty wallet.

Additional simulations on synthetic data based on Gaussian distributions have been performed.

4 Preliminary Results

Among other evaluation criteria, we look at the following: 1. Mean UTXO set size, 2. UTXO set size and composition, 3. mean change per transaction, 4. mean input set size per transaction, and 5. fees paid.

The preliminary results provided in Table 2 show that Bitcoin Wallet for Android (AndroidWallet) achieves the lowest fees and smallest average input set size. Yet, its selection by priority causes the largest measured UTXO set. When high-priority UTXO have been spent, numerous small UTXOs can get selected to create large transactions.

BreadWallet's FIFO approach performs fair in input set sizes, fees and UTXO set size. However, each transaction reveals the maximum age of the wallet's available UTXO.

Compared to BreadWallet, Mycelium's pruned FIFO approach achieves only slightly smaller fees and input set sizes, but more than quadruples the UTXO set size. Similar to AndroidWallet, the build-up of small UTXO at the top of the queue eventually causes transactions with a large input set.

Bitcoin Core performs a complex multi-step Coin Selection procedure requirng a minimum change of 0.01 BTC. As small UTXOs are only introduced by incoming transactions and the knapsack solver selects from the UTXO smaller than the spending target, Bitcoin Core maintains the smallest UTXO set. This comes at the cost of one of the largest total fees, paying 51.6% more than the best performing AndroidWallet.

DoubleWallet, a variation of Bitcoin Core's selection aiming for change of the same size as the spending target, performs worse than Core in almost every category. It has a significantly greater number of UTXOs than Core and increased fees. This is caused by the smaller change outputs created in transactions with small spending targets.

RandomWallet's equi-random selection from all available UTXOs performs very similar to BreadWallet's FIFO approach, maintaining the second smallest average UTXO set.

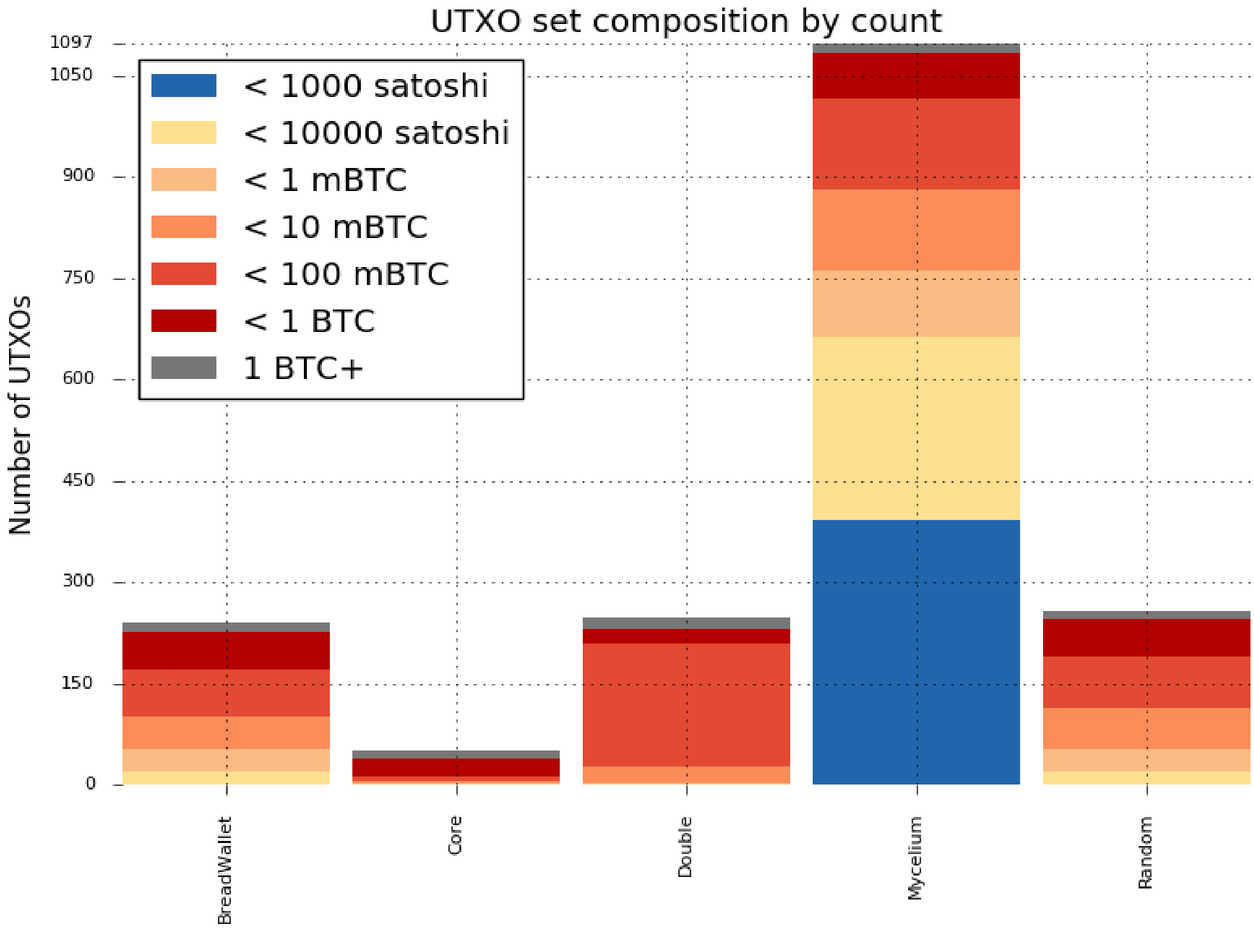
References

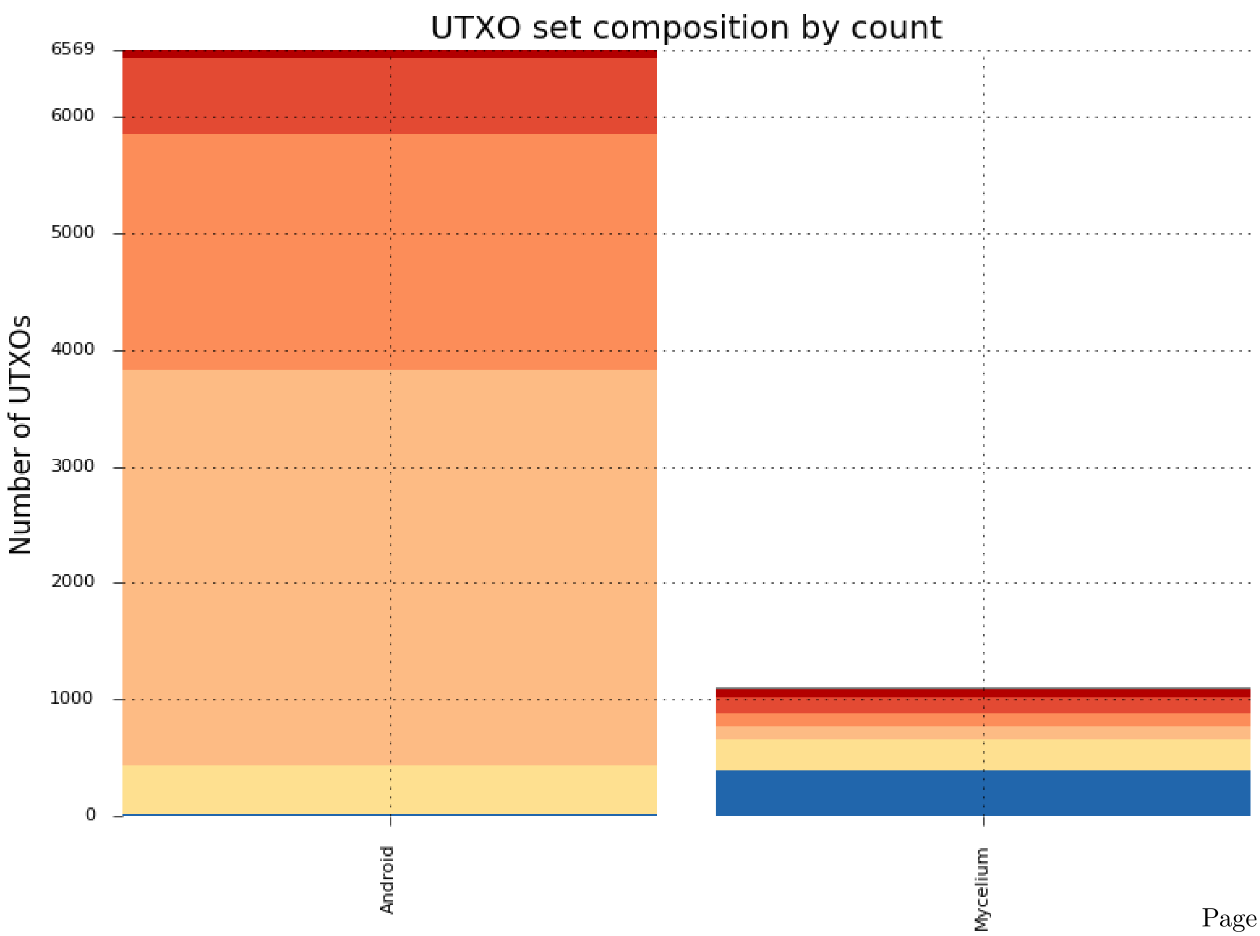
[1] Bitcoin Core developers. Segregated Witness Benefits - Reducing UTXO growth, 2016. [Online, retrieved on 2016-08-26].

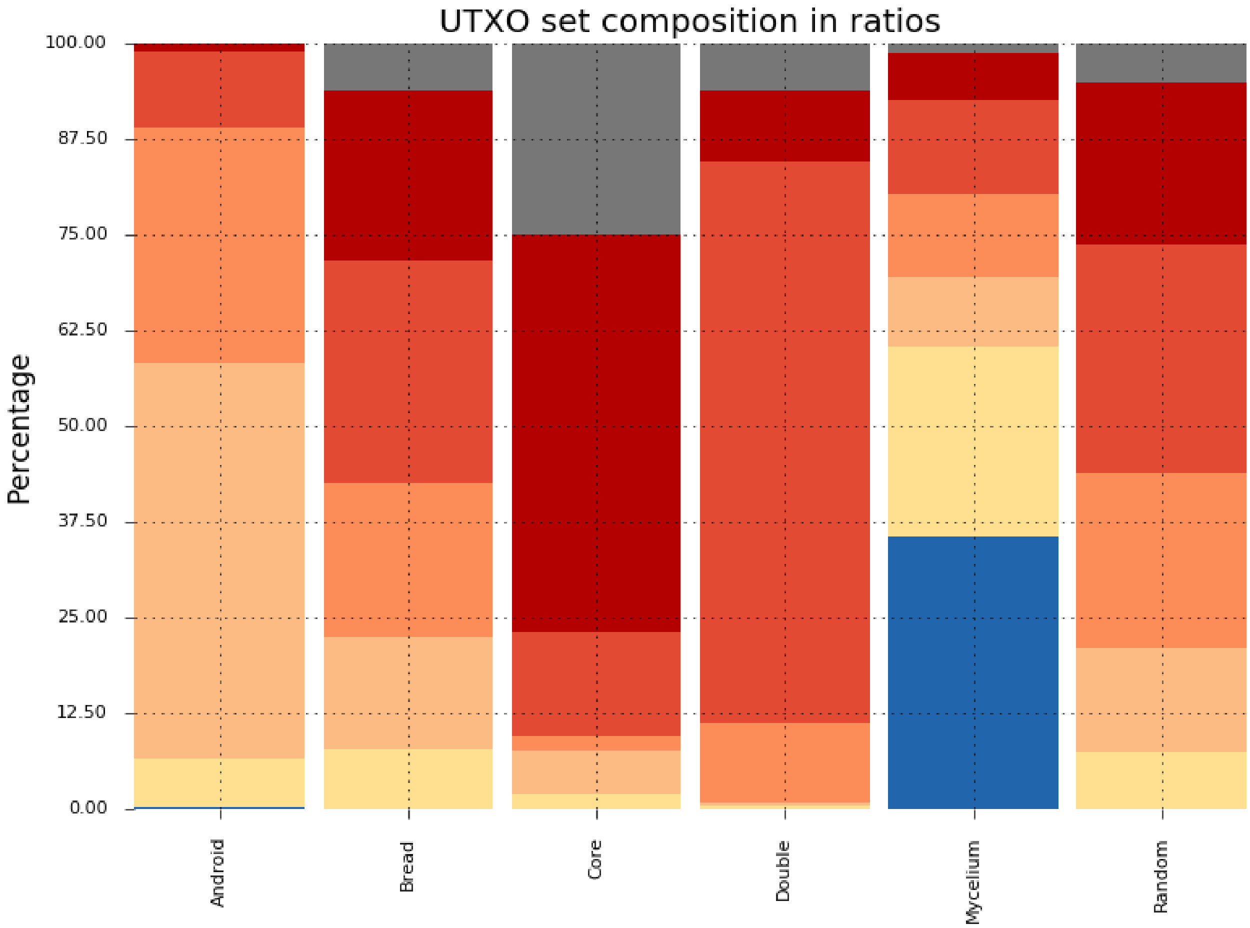
[2] D. Harding. What are the implications of Schnorr signatures?, 2015. [Online, retrieved on 2016-08-26].

[3] R. Havar. Moneypot.com's hot wallet data, 2015. [Online, retrieved on 2016-08-18].

[4] P. Wuille. UTXO breakdown per output amount, 2016. [Online, retrieved on 2016-07-13].


UTXO set composition by count
Number of UTXOs
1097
1050
900
750
600
450
300
150
0
< 1000 satoshi
< 10000 satoshi
< 1 mBTC
< 10 mBTC
< 100 mBTC
< 1 BTC
1 BTC+
BreadWallet
Core
Double
Mycelium
Random


UTXO set composition by count
Number of UTXOs
6569
6000
5000
4000
3000
2000
1000
0
Android
Mycelium


UTXO set composition in ratios
Percentage
100.00
87.50
75.00
62.50
50.00
37.50
25.00
12.50
0.00
Android
Bread
Core
Double
Mycelium
Random